

ОПЫТ И ПРАКТИКА

ПРИМЕНЕНИЕ ПРИНЦИПОВ AGILE В ПРОЕКТАХ ИМПЛЕМЕНТАЦИИ ERP-СИСТЕМ НА ОСНОВЕ КАСКАДНОЙ МЕТОДОЛОГИИ ВНЕДРЕНИЯ

Автоматизация государственных учреждений, реализация стандартов управления в коммерческих компаниях, «Интернет вещей» и цифровизация деятельности предприятий требуют быстрой разработки специализированных приложений. В статье анализируется применимость agile-принципов в ERP-проектах на основе каскадной модели внедрения без перехода к использованию итерационной и спиралевидной схем.

КЛЮЧЕВЫЕ СЛОВА: ERP-системы, agile, водопадная модель, каскадная модель, Scrum, канбан

ВВЕДЕНИЕ

Развитие общества и современных информационных технологий не стоит на месте: каждый год разрабатываются и внедряются десятки государственных программных систем в различных областях жизнедеятельности граждан: производстве, здравоохранении, сельском хозяйстве и пр. Примерами подобных информационных систем служат ЕГАИС (Единая государственная автоматизированная информационная система), ЕГИСЗ (Единая государственная информационная система в сфере здравоохранения), «Госзакупки» и другие решения, интегрированные со множеством прочих программных продуктов сторонних организаций.

В коммерческом секторе имплементируются комплексные программные системы, отвечающие уровню зрелости компании и обеспечивающие ее конкурентоспособность. Внедряются всевозможные приложения, относящиеся к таким стандартам автоматизации, как ERP, SRM, CRM, PLM и SCM [1]. Разработка умных устройств для человека и животных, домов, городов, одежды, транспорта породила независимое направление



Степанов Дмитрий Юрьевич — к. т. н., доцент кафедры корпоративных информационных систем МИРЭА — Российский технологический университет (г. Москва)

автоматизации — «Интернет вещей», согласно которому технические аппараты и веб-приложения, управляющие ими, объединены в единую экосистему посредством сети Интернет.

Все чаще ведутся беседы о необходимости цифровизации деятельности компаний, даже тех, на которых уже внедрены специализированные информационные системы [2], например, автоматизированное сканирование сопроводительных документов (печатные формы ТОРГ-12, счет-фактура), полученных от контрагентов при оприходовании продукции на склад, и их ввод в ERP-систему на основе роботизированных решений.

Все упомянутые выше примеры объединяет потребность в быстрой разработке программного обеспечения. Одной из наиболее популярных и востребованных на сегодняшний день методологий для реализации приложений является гибкая методология agile, которая применяется в итерационной и спиралевидной моделях разработки программного обеспечения. Несмотря на то что идея agile была предложена достаточно давно, ее основные ценности и принципы нашли свое массовое применение лишь в последние несколько лет. В данной статье доказывается применимость agile-принципов ранней поставки продукта, регулярной поставки решения и совместной работы команды заказчика и бизнеса в проектах ERP с использованием водопадной модели.

Демонстрация прототипа заказчику на этапе реализации, миграция данных и их валидация на основе разработанных программ, планомерный перенос разработок в продуктивную ERP-среду, согласование проектной документации и верхнеуровневой архитектуры решения, создание инструкций с использованием готовых приложений обеспечивают сокращение технологических и бизнес-неопределенностей проекта. Несмотря на это agile-методология преимущественно применяется для сопровождения и доработки ранее внедренных информационных систем и требует совершенствования для имплементации в ERP-проекты с нуля и тиражирования.

Цель работы состоит в демонстрации применимости agile-принципов в проектах внедрения ERP-систем, основанных на использовании каскадной модели, для обеспечения наиболее эффективного процесса имплементации.

1. ВОЗМОЖНЫЕ ПУТИ СОЕДИНЕНИЯ КЛАССИЧЕСКОЙ И ГИБКОЙ МОДЕЛИ

Agile-методология применяется в условиях высокой бизнес- и технологической неопределенности [3, 4]. Именно в них итерационная и спиралевидная модели позволяют уточнить бизнес-требования к продукту путем многократного показа предполагаемого решения заказчику. Справедливости ради следует отметить, что подобный подход частично применяется и в водопадной модели, например для уточнения требований в фазе анализа путем прототипирования. Большая часть ERP-проектов на сегодняшний день работают по классической каскадной схеме, по этой причине переход к итерационной или спиралевидной модели потребует значительных организационных и функциональных изменений внутри команды. При этом, как следует из табл. 1, область применения многопроходных моделей достаточно узка [5].

Как же быть в случае высокой неопределенности в проектах внедрения ERP-систем на основе каскадной модели, действительно ли следует переходить к итерационной или спиралевидной схеме? Рассмотрим детальнее три базовых принципа agile-манифеста [6] и их применимость к водопадной модели имплементации информационных систем:

- 1) ранняя поставка продукта;
- 2) регулярные поставки продукта;
- 3) совместная работа представителей бизнеса и проектной команды.

1.1. Ранняя поставка продукта

Ранняя поставка подразумевает регулярную разработку заданной части программы и ее показ

Таблица 1. Целесообразность использования методологии agile в ERP-проектах

| Вид проекта | Способ реализации | Целесообразность | Модель |
|------------------------------------|------------------------|-------------------|--|
| Внедрение с нуля или тиражирование | Настройка и разработка | Минимальная | Каскадная |
| | Настройка | | |
| Развитие системы | Разработка | Средняя / высокая | Итерационная и спиралевидная (методология agile) |

заказчику с целью быстрого выявления возможных ошибок, недочетов и получения комментариев от пользователей. Данный agile-принцип также применим к каскадной схеме с точки зрения:

- уточнения требований при реализации программы;
- миграции основных данных.

ERP-системы обрабатывают огромное количество транзакционных данных, которые отражаются в пользовательских интерфейсах преимущественно в табличном виде, что проще для восприятия и анализа. Архитектура подобных пользовательских программ в ERP-системах едина и состоит из трех экранов [7]: селекционного экрана, позволяющего указывать ограничения для селекции данных, а также экранов данных, содержащих необходимую для обработки информацию и результаты ее обработки (рис. 1).

Все программные разработки в ERP-системах относятся к одному из типов по классификации RICEF, образованной от сокращений: «отчет» (R — report), «интерфейс» (I — interface), «программа обработки данных» (C — conversion), «расширение» (E — enhancement) и «печатная форма» (F — form). Каждая из RICEF-программ имеет заданное число пользовательских экранов (табл. 2). Спецификация для каждой из разработок пары параметров «RICEF-тип и сложность» позволяет унифицировать оценку трудозатрат, необходимых для подготовки функциональной спецификации на разработку, выполнения кодирования и проведения функционально-модульного тестирования программы.

Ранняя демонстрация программы пользователю возможна на этапе реализации и проведения функционально-модульного тестирования, однако

обычно пользователи испытывают программу лишь в процессе приемочного тестирования на завершающем этапе проекта. Структура программ преимущественно задана вне зависимости от вида RICEF, поэтому ранний показ продукта пользователю помогает обсудить удобство использования и эргономику программы, а также критические моменты в работе приложения. Даже этого бывает достаточно, чтобы значительно снизить число замечаний к продукту, не дожидаясь централизованного приемочного испытания.

Вопрос миграции данных ERP-систем является одним из ключевых. Данные принято разделять на основные и переменные в зависимости от частоты их изменения. В литературе часто встречается определение мастер- и транзакционных данных для категорий, указанных выше. Основная сложность миграции заключается в переносе транзакционных данных: они достаточно регулярно обновляются, поэтому возникает необходимость в физической приостановке работы компании для выполнения их копирования в продуктивную ERP-среду.

Ранняя поставка решения заключается в том, что основные данные мигрируют в систему ERP намного раньше ее промышленного запуска. Последнее требует переноса части разработанных программ, например определяющих организационную структуру предприятия, всевозможных справочников и словарей данных, а также прочих объектов, использующихся в качестве атрибутов мастер-данных ERP. Подобный подход предполагает двойной ввод информации в историческую (текущую) и целевую системы в определенный период времени, обычно не больше месяца.

Рис. 1. Пример структуры пользовательской программы в ERP-системе



Преждевременная миграция данных и перенос программ позволяют уменьшить время приостановки работы компании, необходимое для ее подготовки к запуску новой ERP-системы (рис. 2).

1.2. Регулярные поставки продукта

Современные ERP-системы имеют трехуровневый технический ландшафт, состоящий из систем разработки, контроля качества и продуктивной системы. Доработка приложения ведется в системе разработки, технические специалисты и конечные пользователи в среде контроля качества проводят его испытания, и, наконец, оно переносится в продуктивную среду для работы в режиме реального времени.

Проект имплементации ERP-систем подразумевает большое число разработок, доходящих до 500. Массовый перенос приложений в продуктивную среду по результатам завершения приемочного тестирования может обернуться катастрофой: важен порядок переноса программ, программы частично зависят от выполненных настроек ERP-системы, а также могут содержать программный код, не соответствующий правилам наименования, принятым в компании заказчика. Для снижения указанных рисков разработанные и испытанные программы копируются в продуктивную среду по результатам системного и интеграционного тестирования, проводимого намного раньше непрерывного приемочного. Перенос программ требует согласования, именно поэтому результаты испытаний демонстрируются ключевым и конечным пользователям. Таким образом перенос программ планомерно распределяется по времени.

1.3. Совместная работа представителей бизнеса и проектной команды

Внедрение корпоративных информационных систем предполагает достаточно большой объем задач, сгруппированных по этапам и уровням. В «Agile-манифесте» [6] упоминаются базовые составляющие архитектуры любого предприятия: уровни бизнес-процессов, данных, приложений и технической инфраструктуры, дополненные задачами по управлению изменениями и проектом. Каждый из уровней требует тесного

Таблица 2. Виды RICEF-разработок и их пользовательские экраны

| Вид разработки | Селекционный экран | Экран выбранных данных | Экран обработанных данных |
|----------------------------|--------------------|------------------------|---------------------------|
| Отчет | + | + | — |
| Интерфейс | + | + | + |
| Программа обработки данных | + | + | + |
| Расширение | — | — | — |
| Печатная форма | + | + | — |

Рис. 2. Схема миграции основных и переменных данных для запуска ERP-решения



взаимодействия между членами проектной группы исполнителя и заказчика.

Одной из основных причин неудачи проекта имплементации ERP-системы называют незаинтересованность и невовлеченность руководства

и сотрудников предприятия в процесс внедрения. Именно по этой причине двумя ключевыми показателями любого проекта согласно РМВОК [8] являются заинтересованные стороны и план коммуникации с ними. С точки зрения agile-принципа

совместной работы проектной команды и представителей бизнеса можно выделить следующие важные задачи, присущие проекту ERP-внедрения на основе водопадной модели:

- согласование проектной документации;
- миграция переменных данных и тестирование программ;
- подготовка инструкций.

1. *Согласование проектной документации.*

В отличие от итерационной и спиралевидной моделей внедрения каскадная схема подразумевает подготовку пакета документов, включающего проектные решения и спецификации на доработку ERP-системы. Первый документ описывает настройку системы, второй — предполагаемые доработки в разрезе каждого из бизнес-требований. Fit/Gap-анализ, проводимый на этапе проектирования, позволяет понять, каким образом требование будет реализовано в системе: конфигурацией (Fit) или разработкой (Gap), для чего будут подготовлены документы решений и спецификаций.

В проектные решения и функциональные спецификации вначале включают описание исходных бизнес-требований и верхнеуровневой архитектуры, понятной обычному пользователю, и лишь затем технические детали реализации. Таким образом, представитель заказчика, сформулировавший бизнес-требования к системе, еще до реализации программы знакомится с предполагаемым решением на этапе согласования документов. Высказывая замечания к их содержанию, пользователь формирует у себя представление о будущей системе.

2. *Миграция переменных данных и тестирование системы.* Снижение риска загрузки несогласованных данных в продуктивную ERP-систему предполагает выполнение цикла тестовых миграций основных и переменных данных. Процесс миграции состоит из серии итераций, в ходе которых репетируются гармонизация, выгрузка, трансформация, загрузка и непрерывный контроль обработки данных. Каждая последующая итерация характеризуется увеличением объема мигрирующих

данных, таким образом, последняя итерация будет содержать объем, стремящийся к продуктивному.

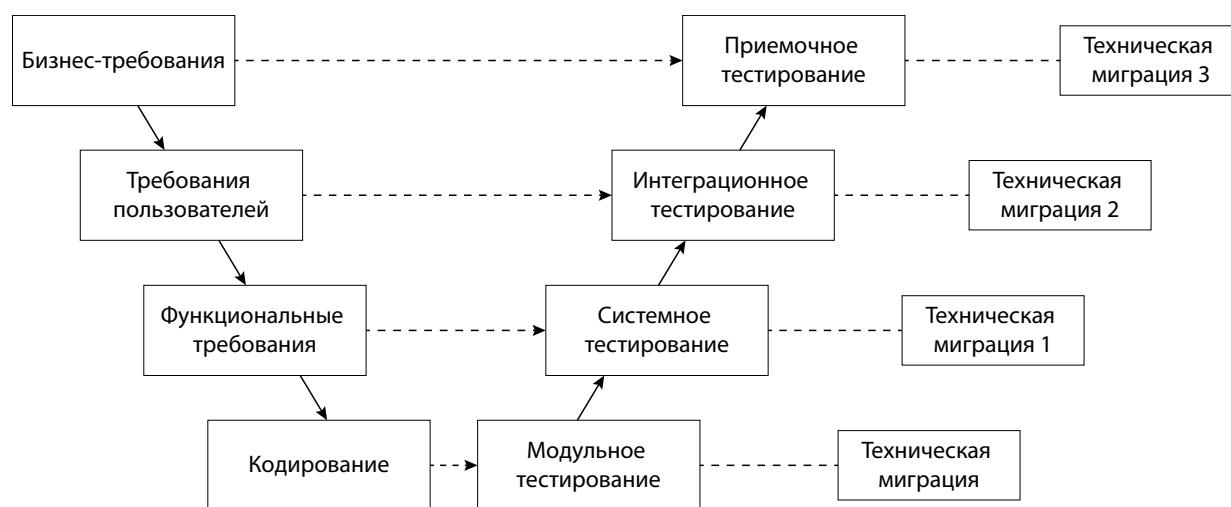
Расширением классической каскадной схемы внедрения информационных систем служит V-модель разработки через тестирование (рис. 3). Особенность этой модели заключается в том, что в ее рамках сопоставляются исходные требования и виды испытаний, в процессе которых проверяется работоспособность решения. Тем самым даже для самых размытых и слабо формализованных требований выполняется проверка реализации в информационной системе.

Обычно результаты тестовых миграций используются как входные данные для проведения тестирования ERP-системы. Более того, загруженные в целевую систему данные должны быть проверены и подтверждены представителями заказчика, для чего в качестве средств валидации и контроля часто используются разработанные программы. Таким образом, циклы тестовой миграции позволяют как наладить работу команды, так и удостовериться в правильности работы приложений для проверки загруженных данных.

3. *Подготовка инструкций.* Обучающие / пользовательские инструкции готовятся в ходе интеграционного тестирования или до этапа выполнения приемочных испытаний. По содержанию документа выделяют бизнес- и функциональные инструкции, каждая из которых включает процессное или операционное описание. Бизнес-инструкция содержит шаги, необходимые для реализации в ERP-системе, а также детальное описание соответствующего бизнес-процесса, относящегося к выполняемым операциям. В отличие от бизнес-инструкции функциональная не имеет как такового описания бизнес-процесса.

Оба вида инструкций могут включать все необходимые транзакции для отражения процесса в системе ERP от начала до конца или лишь одну операцию, в первом случае говорят о процессной инструкции, во втором — об операционной. Вне зависимости от вида инструкция демонстрирует разработанную в ERP-системе программу, а также описывает последовательность шагов

Рис. 3. V-модель разработки через тестирование и циклы миграции данных



для выполнения необходимой операции, что требует глубокого понимания сути разработки. Обычно инструкция создается тем сотрудником, кто изначально формулировал требование к ERP-системе, таким образом осуществляется косвенный анализ соответствия требования и программной реализации в информационной системе.

2. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Существуют три базовые модели внедрения программного обеспечения: каскадная, итерационная и спиралевидная, каждая из них имеет свою область применения. Каскадная схема широко используется в ERP-проектах, где число требований велико и важно выполнить их в полном объеме и в заданный срок. Реестр требований фиксируется и принимается к обработке, возможные флуктуации требований обрабатываются уже как новый проект. Для сохранения лояльности клиента на практике поставщики используют следующий подход: если для реализации нового требования необходимо незначительное число

трудоузатрат, допустим, три человеко-дня, то оно включается в текущий объем проекта без изменения сроков после предварительного согласования.

Итерационные и спиралевидные схемы имплементации на основе agile-методологии используются в случае, когда требования к системе размыты или плохо формализованы, сам заказчик может до конца не понимать, что он хочет получить в итоге. Подобную ситуацию помогает разрешить череда демонстраций полуготовой версии программы, в ходе чего детализируются требования или отсекаются явно не удовлетворяющие клиента программные решения. В отличие от каскадной модели гибкая методология agile допускает и приветствует изменения существующих требований и формирование новых.

Ключевой особенностью каскадной модели внедрения является то, что она обеспечивает минимальную технологическую неопределенность при реализации бизнес-требований. Это достигается за счет того, что кастомизация ERP-решения является фактически одновариантной, а разработка

пользовательских программ выполняется по заранее известной структуре: селекционный экран и экраны выбранных и обработанных данных. Методология agile, разработанная для итерационной и спиралевидной моделей, наоборот, допускает высокую технологическую неопределенность, однако изначально создавалась для обработки ситуации именно бизнес-неопределенности, когда требования настолько поверхностны, что непонятно, каким образом должна выглядеть программа, их реализующая.

Большое число ERP-проектов внедрения с нуля и тиражирования ведутся с использованием водопадной модели имплементации. Такие проекты обладают высокой бизнес-неопределенностью. Возникает вопрос: каким образом возможно снизить подобную неопределенность? Допустимы две опции:

1) переход к использованию гибкой методологии agile, т.е. итерационной или спиралевидной модели внедрения ERP-систем;

2) применение agile-принципов в ERP-проектах на основе каскадной схемы.

Первая опция потребует переобучения проектной команды, изменения методологии ведения проекта и организационной структуры и, самое важное, максимальной вовлеченности бизнеса. Более того, agile-методы имеют достаточно узкую область применения, что показано в табл. 1: проекты сопровождения ERP-систем, требующие программных доработок, но никак не донастроек. Результаты анализа второго варианта рассмотрены в предыдущем разделе и суммированы в табл. 3.

Применение agile-принципов для каскадной модели показало следующее. Большая часть мероприятий, являющихся частью agile, позволяют снизить технологическую неопределенность за счет того, что реализованная программа демонстрируется и применяется намного раньше этапа приемочного тестирования. В частности, результаты системного и интеграционного тестирования программы предоставляются пользователям,

валидация мигрировавших данных проводится с использованием разработанных программ, инструкции готовятся заранее при наличии и возможности запуска уже работающих приложений.

Неопределенности разрешаются следующим образом: предварительное понимание архитектуры программы обсуждается и согласуется с пользователями в момент подписания проектных документов, далее реализованный прототип программы демонстрируется заказчику сразу по завершении функционально-модульных испытаний. С учетом того что каскадная модель является однопроходной и имеет основной недостаток, состоящий в слишком позднем показе решения пользователю, попытка вовлечь заказчика в работу над программой на ранних стадиях проекта (процесс проектирования и модульного тестирования) представляется весьма обоснованной и фактически единственно возможной.

ЗАКЛЮЧЕНИЕ

Несмотря на то что гибкая методология разработки agile часто упоминается в проектах внедрения ERP-систем, на сегодняшний день она находит практическое применение лишь в достаточно узком спектре проектов имплементации информационных систем. В частности, речь идет о сопровождении корпоративных ERP-систем и выполнении их доработки.

Основное преимущество методологии agile заключается в возможности понижения бизнес-неопределенности путем многократной демонстрации работающих прототипов программы заказчику. Проектам внедрения ERP-систем с нуля и тиражирования, реализуемым на основе каскадной модели внедрения, также свойственна бизнес-неопределенность. Одним из способов ее снижения в водопадной модели является применение agile-принципов без перехода к использованию итерационной и спиралевидной моделей.

Таблица 3. Agile-принципы и их применение в каскадной модели внедрения

| Agile-принцип | Задача | Способ решения | Цель |
|--|--|--|---|
| Ранняя поставка | Уточнение требований при реализации программы | Показ прототипа программы на этапе реализации | Снижение бизнес- и технологической неопределенности |
| | Миграция основных данных | Ранний перенос программ в продуктивную среду после проведения приемочных испытаний | |
| Регулярные поставки | — | Планомерный перенос программ в продуктивную среду после проведения системного или интеграционного тестирования | Снижение технологической неопределенности |
| Совместная работа представителей бизнеса и проектной команды | Согласование проектной документации | Описание предполагаемого решения в документах на этапе проектирования | Снижение бизнес- и технологической неопределенности |
| | Миграция переменных данных и тестирование программ | Циклическая репетиция миграции данных и контроль результатов разработанными программами начиная с этапа реализации | |
| | Подготовка инструкций | Документирование деталей работы разработанной программы до проведения приемочного тестирования | |

ЛИТЕРАТУРА

1. Гаврилов Д.А. Управление производством на базе стандарта MRP2. — СПб.: Питер, 2008. — 416 с.
2. Калянов Г.Н. Консалтинг: от бизнес-стратегии к корпоративной информационно-управляющей системе. — М.: Горячая линия — Телеком, 2014. — 210 с.
3. Кон М. Agile: оценка и планирование проектов / Пер. с англ. — М.: Альпина Паблишер, 2019. — 418 с.
4. Laufer A. (1996). *Simultaneous Management: Managing Projects in a Dynamic Environment*. New York: AMACOM, 313 p.
5. Степанов Д.Ю., Вельсовский А.В. Применение Agile Scrum в проектах SAP // Корпоративные информационные системы. — 2018. — №1. — С. 9–17.
6. основополагающие принципы agile-манифеста. — <http://agilemanifesto.org/iso/ru/principles.html>.
7. Степанов Д.Ю. Формирование универсальных требований к пользовательским программам при подготовке спецификации на АВАР-разработку // Актуальные проблемы современной науки. — 2014. — Т. 78. — №4. — С. 258–268.
8. Руководство к своду знаний по управлению проектами. — 5-е изд. — М.: Олимп-Бизнес, 2014. — 586 с.